

**TB-0364**

## **COBOL REPORT WRITER UPDATE**

Issue Date: July 12, 2005  
Effective Date: July 19, 2005  
Section/Group: Mainframe Systems  
Submitted by: Dennis Hansen  
Approved by: Clair Christensen

---

This entire document is of importance to all COBOL and COBOL Report Writer users.

COBOL Report Writer V1R4M0 has been updated to accommodate the new version of Enterprise COBOL, including APARS VJR4087, VJR4088, VJR4089, VJR4090, VJR4091, and VJR4092. These updates will be implemented on CPUs 5 and 6 on Tuesday, July 19th, 2005 and on CPUs 2, 3, 4, 7, and 9 on Thursday, July 21st, 2005. These updates correct the following problems:

### **VJR4087—Problem Description**

If the precompiler is invoked by means of an attached macro, any list of alternate ddnames given as parameter 2 were ignored. For example, if IBM sclm is used to control invocation of the precompiler, its ddname substitution does not work. This fault does not apply to the compiler invoking the precompiler via an inexit and does not apply to invocation via a link macro, both of which work correctly.

- This update is valid only if LE/370 is used in the STEPLIB (or in the link list) during execution of the pre-compiler.
- It may, in the future, be superseded by a more general update.
- This apar is an updated version of vjr4070, for applying after ptf uj04010 has been installed.

### **VJR4088—Problem Description**

The precompiler fails if your customized le/370 settings specify 'all31(on)'. The default setting is all31(off), which means that some programs may run with an addressing mode (amode) of 24, typical of the "legacy" situation. However, it is becoming common for sites to change this default to all31(on) in the expectation that all programs should run with an addressing mode of 31, unless overridden by a run-time option. This is



efficient in many scenarios because le/370 can then pre-assign a block of memory from higher storage in the knowledge that every program will be able to access it.

Unfortunately, certain programs in the precompiler load library have their attributes set to amode(24), in spite of the fact that they are capable of running with amode (31). In the all31(off) situation, this is not a problem. But in the case of all31(on), these programs fail with an "abend" unless the option is changed for the precompile step to all31(off). This is not possible if the precompiler is called as an "inexit" by the compiler because the compiler does not accept le/370 run-time parameter options. For example, parm=".../all31(off)" is often used in an application program but is not valid with the compiler program.

This update changes the attributes of the affected precompiler programs to amode(any).

### **Vjr4089—Problem Description**

If the option cics is coded using the cbl/process statement at the start of a source, the precompiler does not pass it on to the compiler, even if the \*\*control norw option is used. Also, if the cics('string') format is used, the precompiler generates an error message.

### **Vjr4090—Problem Description**

The "xml generate" statement of IBM Enterprise COBOL is incorrectly intercepted by the precompiler because of the "Generate" keyword. This amendment cures this problem.

### **VJR4091—Problem Description**

If the input source contains certain non-graphic characters, in particular a hex "27", the precompiler may operate incorrectly.

### **VJR4092—Problem Description**

If the compiler "list" (assembler code) option is used with the rw inexit and prtexit and the compiler is Enterprise COBOL (IBM COBOL 3.1 onwards), then the print exit copies the assembler listing without altering the source line numbers to correspond to the original source. This causes debug operations to fail when the assembler listing cannot be matched to the procedure division shown in the same listing.



### PLEASE NOTE:

There are many OLD libraries in existence that we would like to clean up and get every one using the same libraries. There is a lot of duplication of program libraries so we are at a point where we don't know what is what.

VS COBOL, or OS/VS COBOL as it is usually called, was IBM's ANSI-68 COBOL offering, dating back to punched-card days.

VS COBOL II was the first of an entirely new breed and COBOL Report Writer runs with them all. So, as it turns out, only one proc 'could' be used to pick up VS COBOL II, COBOL for MVS or Enterprise COBOL. So, from a technical level, they look and behave like the same animal.

The current version of Language Environment (LE) will service all Cobols., i.e. it contains all the callable routines needed to satisfy references generated by any current compiler (and most obsolete ones).

For that reason we would like to propose the following:

The following libraries will be renamed and saved (just in case) for six months:

DP.COBRW.SCXRCOBA  
DP.COBRW.SCXRCOBQ  
DP.COBRW.SCXRJCL  
DP.COBRW.SCXRPREC  
DP.COBRW.SCXRRUN

DP.COBMVS.SEQACLIS  
DP.COBMVS.SEQAIENU  
DP.COBMVS.SEQAMOD  
DP.COBMVS.SEQAOS2  
DP.COBMVS.SEQAPROC  
DP.COBMVS.SEQASAMP  
DP.COBMVS.SEQA2ENU  
DP.COBMVS.SIGYCLST  
DP.COBMVS.SIGYCOMP  
DP.COBMVS.SIGYMAC  
DP.COBMVS.SIGYPROC  
DP.COBMVS.SIGYSAMP

GP.COB2RWTR.COMP  
GP.COB2RWTR.LIB



GP.COB2RWTR.PROD.COBQRLIB  
GP.COB2RWTR.PROD.PRECLIB  
GP.COB2RWTR.PROD.RUNLIB  
GP.COB2RWTR.TABLE

SYS1.COB2CICS  
SYS1.COB2CLIB  
SYS1.COB2COMP  
SYS1.COB2LIB  
SYS1.COB2MLIB  
SYS1.COB2PLIB

SYSX.COB2RWTR.LIB

DP.SPC.V1R4M0.APARS  
DP.SPC.V1R4M0.JCLLIB  
DP.SPC.V1R4M0.PTFS  
DP.SPC.V1R4M0.SCXRCOBA  
DP.SPC.V1R4M0.SCXRCOBQ  
DP.SPC.V1R4M0.SCXRJCL  
DP.SPC.V1R4M0.SCXRPREC  
DP.SPC.V1R4M0.SCXRRUN  
DP.SPC.V1R4M0.SCXRRUN.Y  
DP.SPC.V1R4M0.TCXRJCL  
DP.SPC.V1R4M0.TCXRJCL.Y  
DP.SPC.V1R4M0.TCXROBJ  
DP.SPC.V1R4M0.TCXRPREC  
DP.SPC.V1R4M0.TCXRRUN

The following libraries will be copied to renamed libraries and the contents of these libraries will be replaced with the updated version of Cobol Report Writer.

The exception to this is the SYS1.COBRW.SCXRJCL library which will be replaced with SYS1.COBRW.SAMPPROC. The SAMPPROC library will have sample JCL that can be referred to by all users to set up their own JCL for executing the Cobol Report Writer.

SYS1.COBRW.SCXRCOBA  
SYS1.COBRW.SCXRCOBQ  
SYS1.COBRW.SCXRJCL  
SYS1.COBRW.SCXRPREC  
SYS1.COBRW.SCXRRUN



The following two(2) libraries appear to be the VERY, VERY old VS Cobol. These libraries will NOT be updated, copied nor changed:

SYS1.VSCLLIB  
SYS1.VSCOLIB

The current and latest version of the Enterprise Cobol is contained in the following libraries:

SYS1.IGY.SIGYCLST  
SYS1.IGY.SIGYCOMP  
SYS1.IGY.SIGYMAC  
SYS1.IGY.SIGYPROC  
SYS1.IGY.SIGYSAMP

The following Language Environment (LE) data sets should be compatible with most versions of Cobol and Cobol Report Writer:

SYS1.CEE.SAFHFORT  
SYS1.CEE.SCEEBIND  
SYS1.CEE.SCEEBND2  
SYS1.CEE.SCEECICS  
SYS1.CEE.SCEECLST  
SYS1.CEE.SCEECPMAP  
SYS1.CEE.SCEECP  
SYS1.CEE.SCEEGXLT  
SYS1.CEE.SCEEH  
SYS1.CEE.SCEEH.ARPA.H  
SYS1.CEE.SCEEH.H  
SYS1.CEE.SCEEH.NET.H  
SYS1.CEE.SCEEH.NETINET.H  
SYS1.CEE.SCEEH.SYS.H  
SYS1.CEE.SCEEH.T  
SYS1.CEE.SCEELIB  
SYS1.CEE.SCEELKED  
SYS1.CEE.SCEELKEX  
SYS1.CEE.SCEELOCL  
SYS1.CEE.SCEELOCKX  
SYS1.CEE.SCEELPA  
SYS1.CEE.SCEEMAC  
SYS1.CEE.SCEEMSGP



SYS1.CEE.SCEE OBJ  
SYS1.CEE.SCEEPROC  
SYS1.CEE.SCEE RUN  
SYS1.CEE.SCEE RUN2  
SYS1.CEE.SCEESAMP  
SYS1.CEE.SCEESPC  
SYS1.CEE.SCEESPCO  
SYS1.CEE.SCEEUMAP  
SYS1.CEE.SCEEUTBL  
SYS1.CEE.SIBMAM24  
SYS1.CEE.SIBMCALL  
SYS1.CEE.SIBMCAL2  
SYS1.CEE.SIBMMATH  
SYS1.CEE.SIBMTASK

If you desire to test with the new updated version of Cobol Report Writer, you can access them, (FOR NOW), by referring to the following names:

DP.COBRW.V1R4M0.SAMPPROC  
DP.COBRW.V1R4M0.SCXR COBA  
DP.COBRW.V1R4M0.SCXR COBQ  
DP.COBRW.V1R4M0.SCXRPREC  
DP.COBRW.V1R4M0.SCXR RUN

Hopefully, this will clean up all the libraries found out on the systems. If you find any others, or you have any questions, please call me or email me and I will be happy to help and take another look.

Thanks,

Dennis Hansen  
(801)538-3504  
[dhansen@utah.gov](mailto:dhansen@utah.gov)

